Life (and routing) on the Wireless Manifold

Varun Kanade* Santosh Vempala†
College of Computing, Georgia Tech
{varunk,vempala}@cc.gatech.edu

ABSTRACT

We present the *wireless manifold*, a 2-dimensional surface in 3-dimensional space with the property that geodesic distances accurately capture wireless signal strengths. A compact representation of the manifold can be reconstructed from a sparse set of signal measurements. The manifold distance suggests a simple routing algorithm that avoids obstacles, naturally handles mobile nodes without explicitly maintaining the connectivity graph and is more efficient compared to using Euclidean distance as measured by success rate, routing load and failure tolerance. Placing sensors to cover the manifold is more effective than covering the underlying physical space.

1. Introduction

The connectivity graph of a wireless network is determined by complex factors such as geographic layout, physical obstacles, noise and electromagnetic interference. Moreover, a principal feature of ad hoc networks is the mobility afforded to the nodes and this implies that the topology of the network could be frequently changing.

A widely studied model for wireless connectivity is the unit disk graph model [8]. Each node of the network is represented as a point on the 2-dimensional plane and two nodes are connected if their distance is at most 1, i.e., the range of a node is a unit circle centered at its location. While this model is attractive and often used for algorithm design or validation, it does not take into account the many factors other than distance that affect connectivity and indeed the disk assumption is often violated in practice [2, 13, 12].

In this paper, we define the *wireless manifold* using the basic principle that signal strength decay follows an inverse square law. The key property of the wireless manifold is that the shortest path (geodesic) distance along the manifold between any two points estimates the signal decay between them. Thus, the connectivity graph is determined by disks *on the manifold*; a disk of radius r on the manifold is the set of all points within geodesic distance r. The contours of these disks on the plane may be very different from a circle depending on the structure of the manifold at the point.

We give an algorithm to construct a compact representation of the manifold from a sparse set of signal strength measurements. Without directly modeling any physical factors, the representation captures and predicts signal decay to very high accuracy and is vastly superior to the best possible Euclidean (planar) representation, thus improving on the unit disk model and its known refinements. It is conceptually different from previous work on assigning virtual coordinates to the nodes in a network [5, 20, 3] or modeling non-Euclidean features of network connectivity by explicitly modeling obstructions [4]. The manifold representation makes obstructions implicit, making the model conceptually much simpler and yet accurate.

Our representation decomposes the connectivity graph of a wireless network into two parts — the manifold itself which does not change often and the locations of nodes which might change frequently but are easy to update. This can be used in several scenarios; two important ones are (a) routing and (b) sensor placement.

Routing. Geographic routing [6, 14, 10, 16, 15] is an elegant approach to wireless communication. It requires planarization of graphs by techniques such as the Relative Neighborhood Graph (RNG)[21] and the Gabriel Graph (GG) [7]. Kuhn *et al.*[17] have proposed relaxing the unit-disk graph assumption to improve robustness of planarization techniques which fail in case the underlying graph violates the unit disk assumption. Kim *et al.*[12] propose the cross link detection protocol, which enables provably correct geographic routing on arbitrary connectivity graphs. All these approaches involve deleting certain links to guarantee success of routing algorithms by avoiding holes.

We use the manifold to implement a simple routing algorithm without removing any links. The idea is that between two points, there exists a path on the manifold that is monotonically decreasing in distance, as the manifold is raised in areas where there are obstructions and is flat elsewhere. The manifold is represented as an edge-weighted grid which makes it easy to maintain shortest path distances. For routing, each packet has a header indicating its current manifold distance to the target node. Any node that receives the packet retransmits if and only if its distance to the target is less than the current distance of the packet (in the header) by a certain *width* parameter (which can be tuned to reduce network load) and updates the header. We present experimental results demonstrating the main properties of the algorithm.

When a node moves to another location, it has to update its location on the manifold and inform other

*Supported in part by ARC ThinkTank and by the NSF.
†Supported in part by the NSF. This article is Georgia Tech CS Tech Report GT-CS-07-06

nodes that wish to send to it. We can implement known location service methods such as [18] for this purpose. The manifold representation is suited for mobility since the manifold itself is not expected to change rapidly (e.g., new buildings). All that needs to be maintained are the node locations. Unlike many previous routing algorithms manifold routing does not rely directly on the connectivity graph. Even though the connectivity graph could be rapidly changing in response to node movements, it is implicitly known from the wireless manifold (which is more stable) and the node locations. Further, when the manifold does change, our algorithm for finding a representation can also be used to refine it.

In contrast with ad hoc on-demand distance vector (AODV) [19] routing or dynamic source routing (DSR) [9] the manifold routing algorithm does not incur overheads of route discovery or of carrying the entire route in the header. However by tuning the width parameter, we demonstrate that it is possible to ensure fairly low routing load on the network, as is the case with these algorithms.

Placing sensors. The main goal of sensor placement is to provide coverage of a certain area or connectivity (in the case of relays). Placing nodes uniformly on the underlying physical space ignores obstacles and interference. On the other hand, the wireless manifold incorporates all these features and is a better map for choosing sensor locations. We present an algorithm that using a few measurements learns the manifold, and this identifies regions where the connectivity needs to boosted. We demonstrate that choosing sensor locations to maximize coverage on the wireless manifold can be considerably superior to choosing them to cover the plane.

2. The wireless manifold

In this section, we describe a simplified model for manifolds that will suffice for our purpose. Our manifolds will be distorted 2-dimensional grids. More precisely, let V be the set of grid points of a 2-dimensional $k \times k$ grid with coordinates from the set $\{1, \ldots, k\}$ and E be the set of all grid edges, i.e., pairs of adjacent vertices on the grid. A manifold M is obtained by assigning a nonnegative length, $l(i,j)$, to each grid edge $(i,j) \in E$. These lengths induce a metric on the grid vertices where the distance $M(u,v)$ between each pair of grid points u, v is the length of the shortest path. The set of all manifolds, $\mathcal{M}$, is the set of all such metrics induced by length assignments to the grid edges.

We now define the *best manifold* problem. The input is the set of locations of wireless nodes and the measured signal strengths for some pairs of nodes, i.e., a subset W of the grid vertices U and a subset F of pairs of points from W along with nonnegative "distances" $d(u,v)$ for each pair in F. The problem is to find the manifold M, i.e., lengths $l(i,j)$ for grid edges, so that the induced shortest path metric is as close as possible to the given distances for pairs in F in the following measure:

$$\min_{M \in \mathcal{M}} \sum_{(u,v) \in F} \left(d(u,v) - M(u,v)\right)^2 . \qquad \text{(I)}$$

Alternative formulations are possible, e.g., one could restrict the final metric to be non-contracting, i.e., $M(u,v) \geq d(u,v)$, and then minimize the maximum distortion, i.e., minimize the maximum of $M(u,v)/d(u,v)$ over pairs in F. Another possibility is to minimize the average distortion.

Unfortunately, all these formulations are NP-hard, even to approximate. This follows due to a reduction from planar 3-SAT (we omit the proof in this paper). Thus, we do not hope to find an efficient algorithm to solve the minimization problem in the worst-case.

On the other hand, the objective function (I) is differentiable, and we use the following method to find a local optimum.

1. Set every edge to the same length (we use the value that minimizes (I)).
2. Repeat: Compute the gradient of (I) at the current point and move in the direction of the gradient without leaving the positive orthant; if the gradient makes some edge negative, project the gradient to the positive orthant and move in that direction.

3. Sensor placement

Once we have identified a manifold, we can use it to guide the placement of network resources such as sensors or relays.

We place nodes so as to cover the wireless manifold rather than the Euclidean plane on which the nodes lie. In a random placement, we choose each grid point with probability proportional to the sum of the lengths of adjacent edges. In Section 5, we compare uniformly picking grid nodes (which corresponds to sampling the Euclidean plane) to picking them according to the manifold, to see which method achieves connectivity and coverage faster.

Besides connectivity, placing relays using the manifold can also be advantageous for routing. We discuss and evaluate that aspect in the context of the routing algorithm.

4. Routing

The basic idea for routing is very simple. Imagine each node has a table of the distances to every other node in the network (we will shortly see that we do not need an explicit table and distances can be computed from a small representation). A packet P has two pieces of information in its header: (1) $Dest(P)$, the identity of its final destination, and (2) $Dist(P)$, a number denoting the distance from the location of its most recent retransmission to $Dest(P)$. The distance used is the manifold distance, $M(.,.)$. Let $R(v)$ denote the radius of influence of node v. Forwarding is done according to the following rule:

When node v receives packet P,
– if $M(v, Dest(P)) < Dist(P) - \alpha R(v)$,
– set $Dist(P) := M(v, Dest(P))$ and retransmit P.

We also allow nodes to broadcast. This is done by setting the destination ID to a special character. Broadcast is useful, e.g., when a node has to announce its new location.

The *width* parameter α can be tuned for efficiency. At $\alpha = 0$, the algorithm is guaranteed to find a route if one exists but might use a larger number of nodes for retransmission than necessary. As we increase α, the set of nodes participating reduces. We propose that α be adjusted dynamically, similar to congestion control in wired networks. Figure 7 in the evaluation illustrates contours of network load for different values of α on a randomly generated instance. As we point out in the discussion, each node could have its own width setting.

We view the size of the grid to represent the manifold as a constant independent of the size of the network. Each node keeps a copy of the grid. The distance between two network nodes is estimated by the distance between their nearest grid points.

5. Evaluation

In this section, we present our preliminary evaluation for (a) learning the manifold (b) sensor placement and (c) routing efficiency and failure tolerance.

We report experiments on the CRAWDAD data set Rutgers-noise [11] and randomly generated instances. The Rutgers-noise data sets contain RSSI (Received Signal Strength Indication) measurements for some pairs of nodes. The data set was collected by injecting noise into the ORBIT[1] indoor testbed, which consists of an 8×8 grid, with nodes chosen at some locations. There are 29 nodes, each node acts as a transmitter and RSSI measurements are made at receiving nodes. The wireless signal is affected by the noise injected in the system. We used data at 3 noise levels 0 dB(dbm 0), -5 dB(dbm-5) and -20dB(dbm-20).

We also randomly generated manifolds for evaluation. For this we used Gaussian distributions with randomly chosen centers and random covariances. Each Gaussian affects the lengths of manifold edges that are at least some distance away from its center. The length of an edge is defined by the Gaussian that places the maximum density at the midpoint of the edge (see Fig. 4 for an example). We chose this model because it captures the effects of having steep barriers which may be caused by obstructions.

5.1 Finding the manifold

We apply the algorithm described in Section 2 to each of the Rutgers-noise data sets. The measured RSSI values are converted to distances which are input to the algorithm.

We evaluated the quality of the recovered manifold, i.e., how closely it approximates original values and also how well it generalizes. We do this by randomly omitting a subset of the observed measurements and then comparing them with the values predicted by the manifold representation found on the remaining observations.

Figure 1 shows the manifold obtained by the algorithm for the data set dbm0. The red dots are positions of network nodes. The shortest path between a source – destination pair is marked. It can be seen that shortest paths on the manifold can be very different from shortest Manhattan or Euclidean paths. The manifold also gives an idea of barriers across which communication is harder and thus suggests points to place relays.

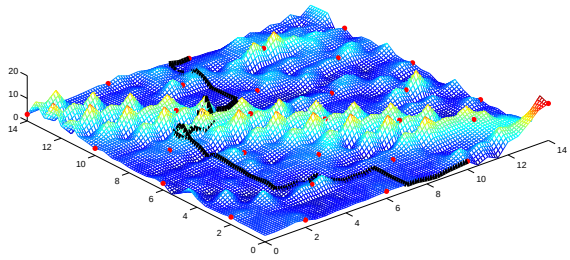

Figure 1: : Manifold for dbm0

Data set	Measure	Euclidean	Manifold	Manifold prediction
dbm0	Avg Error	0.185	0.017	0.07
	Max Dist	25.85	2.86	1.71
	Avg Exp	2.33	1.13	1.25
	Avg Contr	0.35	0.9	0.93
dbm-5	Avg Error	0.137	0.022	0.03
	Max Dist	14.81	2.58	1.87
	Avg Exp	1.83	1.15	1.36
	Avg Contr	0.44	0.89	0.87
dbm-20	Avg Error	0.145	0.021	0.02
	Max Dist	14	2.13	1.79
	Avg Exp	1.72	1.11	1.12
	Avg Contr	0.29	0.88	0.89

Table 1: Embedding data sets into manifolds

To evaluate how good the manifolds are we use several measures. Table 1 compares the manifold with the best plane embedding in terms of (1) average error: the ratio of the objective function value (I) to the sum of squares of all the distances in the graph, (2) maximum distortion: the product of maximum contraction (factor by which an edge length is reduced) and maximum expansion (factor by which an edge length in increased), (3) average expansion: the average amount by which distances were stretched up compared to the original, and similarly (4) average contraction. From the first and second columns of the table, we see that the manifold embedding typically has a 5 to 10-fold advantage over

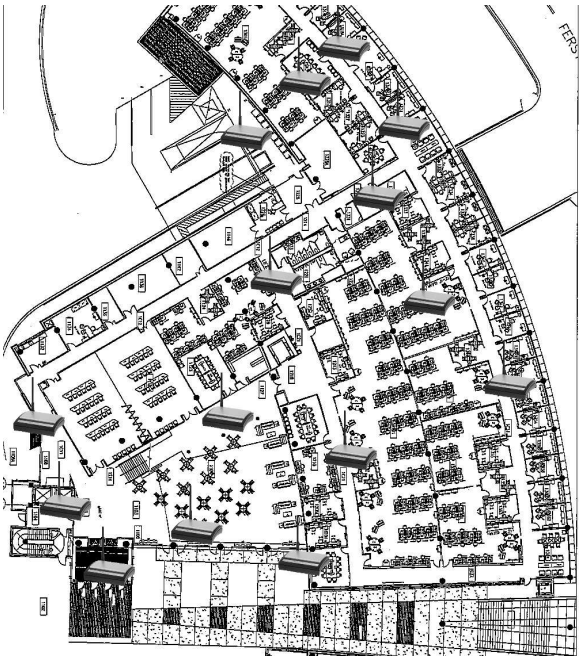

Figure 2: Placement of nodes in Klaus building at Georgia Tech

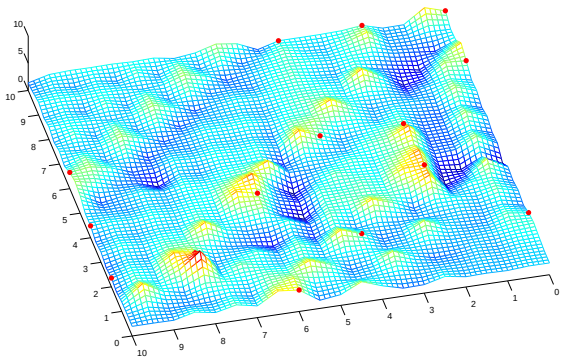

Figure 3: Manifold for the Klaus building at Georgia Tech

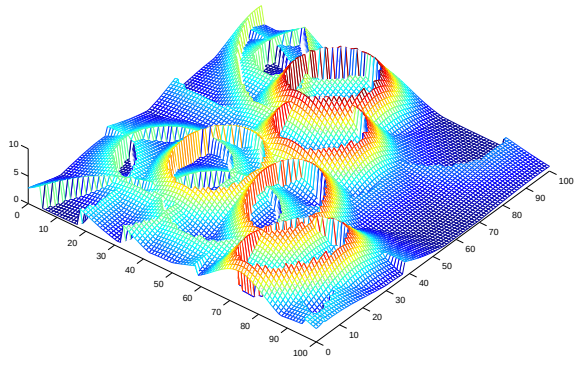

Figure 4: Randomly generated manifold

Euclidean and further the actual error values are quite small. Further, the average error of the rank-5 approximations for the three datasets dbm0, dbm-5, dbm-20 were 0.093, 0.086 and 0.066 respectively. The matrix of specified distances does not have a good low-rank approximation, implying that even using more coordinates does not help if we restrict ourselves to Euclidean distance. The curvature induced by the manifold is essential.

To test how well the learned manifold generalizes, we dropped at random 8% of the known distances (edges) from the data sets, computed the manifold on the rest of the observations and made preidctions on the dropped 8% using the computed manifold. The last column in Table 1 shows that the manifold prediction error is low on all the measures and is comparable to that on the full set of values. From this we conclude that the manifold captures and generalizes wireless connectivity accurately.

We collected signal strength data by placing wireless nodes at various positions on the first floor of the Klaus Advanced Computing Building at Georgia Tech. Figure 2 shows the floor plan of the building with the placement of the nodes. Figure 3 shows the manifold obtained for this region. The average error, expansion and contraction were again significantly smaller for the manifold compared to the best Euclidean scaling.

5.2 Sensor placement

We used randomly generated manifolds for this section. Figure 4 gives an example. For a fixed radius r (two grid points on the manifold can communicate directly if their manifold distance is at most r) we choose node locations on the manifold in two ways: (i) choose grid points uniformly (ii) choose grid points with probability proportional to the sum of the incident edge lengths as defined by the manifold. We then report the number of nodes required to be chosen to ensure that the network is connected. Figure 5 show the plot of the number of nodes required to achieve connectivity vs the radius of transmission for the two different ways of choosing sensor locations.

We also measured the number of nodes for full *coverage* vs the radius and separately, the fraction of pairs connected vs number of nodes and fraction of the grid covered vs number of nodes. In every case, manifold sampling was significantly better.

5.3 Routing

We use the following measures for evaluating the routing algorithm in simulations: (1) Success rate, i.e. fraction of node pairs that can communicate (2) Routing load, i.e., the average number of packets forwarded in

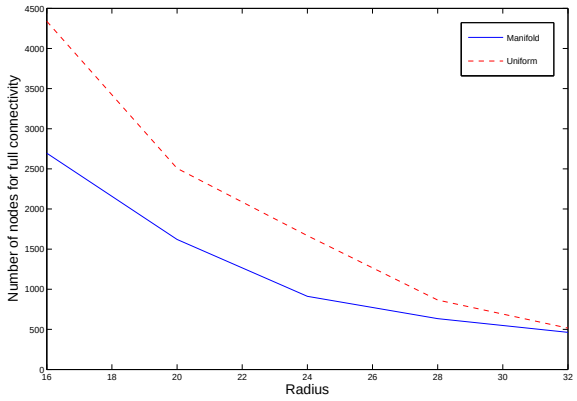

Figure 5: Number of nodes required to achieve perfect connectivity at different radii

the network per node. We implement the algorithm by using the manifold distance and also (for comparison) Euclidean distance. The latter is related to greedy geographic routing. We compare these methods on each of the Rutgers-noise data sets as well as on random manifolds. In the former case, we routed between every pair of nodes. For randomly generated manifolds, we picked network node locations in two ways. First, we picked every other grid point vertically and horizontally and routed between random pairs. Second, we picked a set of N nodes, by picking $N/2$ random grid points and the other $N/2$ either at random from the manifold or uniformly from the grid, corresponding to the routing algorithm used. Messages were routed between random pairs drawn from the common nodes (the first $N/2$). The motivation for the last choice was to study the effect on routing of choosing relay nodes in the two different ways.

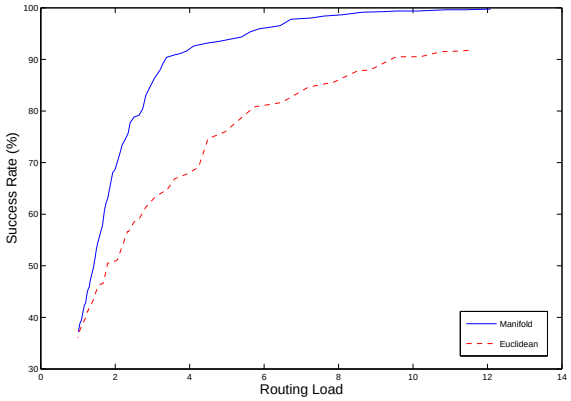

Figure 6: Routing on dbm0

On each of the three Rutgers-noise data sets, at every radius of transmission that we used, manifold routing had a higher success rate and smaller load. The routing load can be adjusted by varying the width parameter α of the routing algorithm. Figure 6 plots the success rate vs the average load for the dbm0 data set. We see that to ensure 90% success rate, manifold routing incurs a routing load of less than 4, where as Euclidean routing incurs a routing load of 11.5, nearly 3 times as much. We found similar plots for the other two data sets.

We next report results of routing on randomly generated manifolds. Figure 7 shows the nodes retransmitting packets while attempting to route between the source (left hand bottom star) and destination (right hand top star). Manifold routing is successful for α up to 0.5. In the figure, the nodes marked in white circles are the only ones retransmitting at $\alpha = 0.5$ and as α goes to zero, the darker ones also start retransmitting and finally at $\alpha = 0.0$, all marked nodes are retransmitting. Euclidean routing tends to flood almost the entire grid when it succeeds.

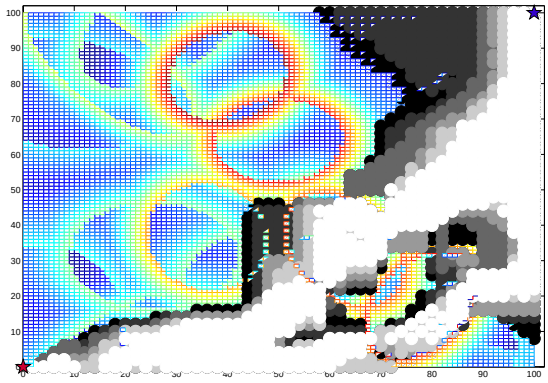

Figure 7: Manifold routing

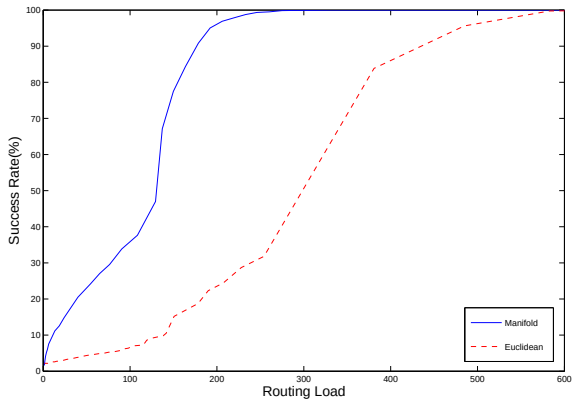

Figure 8: Routing on a random manifold

Figure 8 shows the trade-off between success rate and routing load for the two routing methods on the first experiment with random data using a subgrid of points as

node locations. We see that to ensure 95% success rate, manifold routing incurs a routing load of 178, whereas for the same success rate Euclidean routing incurs a routing load of 485. In the second experiment, Figure 9 shows the same comparison for the randomly chosen network node locations. To achieve 95% success rate manifold routing incurs routing load of 265 whereas for the same success rate, Euclidean routing incurs a routing load of 603. Thus by tuning the width parameter α, we see that a large reduction in network load is possible for manifold routing without reducing the success rate significantly.

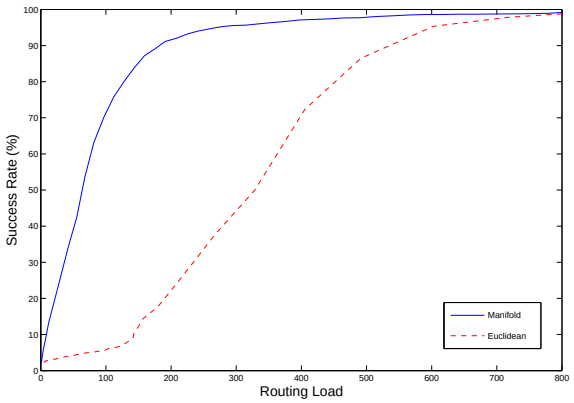

Figure 9: Routing with randomly chosen node locations

Finally, we also ran the routing experiments on random manifolds after failing each node independently with probability p. For p from 0 to 0.8, the success rate using manifold distance was always higher than that using Euclidean distance.

6. Discussion and Future Work

Our results on recovering the wireless manifold and predicting signal strengths using it provide compelling evidence that (a) the manifold captures wireless communication with high accuracy without explicitly modeling complex physical factors (b) the manifold can be recovered using a simple algorithm. We plan to investigate these findings more thoroughly in different test zones including large buildings and urban landscapes. We will also study the size of the manifold representation required to produce an accurate distance measure and nonuniform grids, e.g., allowing the grid to be finer in places.

From our preliminary experiments, the routing algorithm based on the manifold distance appears to be quite effective and can be tuned for routing load efficiency, e.g., we can allow individual nodes to set their own values of the width parameter α. The manifold representation provides a solution to constantly changing mobile topologies, by decoupling node locations from the characteristics of the terrain. However, so far we have only tested the routing algorithm in simulation. We plan to fully evaluate the algorithm by setting up a multi-hop wireless network and measure throughput, latency and recovery time under failures and mobility. Further, it appears likely that the representation will provide improvements for other distance-based routing algorithms, by using manifold distance.

An exciting scenario for multi-hop wireless networks is providing connectivity in developing countries that face the "last-mile" problem. Broadband access is expensive and while fiber is available it comes within a mile of most users but not all the way. In ongoing work with the TeNet group in Chennai, India, we will study manifold routing along with other known approaches in an urban area that we have already identified. This region is particularly interesting because other solutions such as building powerful antennas or providing cable access are prohibitively expensive or otherwise impractical. Further, power consumption is one of the considerations and so multi-hop and low network load can both play a useful role.

Finally, our results raise some interesting theoretical questions. In particular, it would be nice to characterize the set of distance metrics that are approximately embeddable in 2-dimensional manifolds.

Acknowledgements. We are grateful to Ellen Zegura, Anirudh Ramachandran and Nick Feamster for helpful comments.

REFERENCES

[1] `http://www.orbit-lab.org/`.

[2] L. Barriére, P. Fraigniaud, and L. Narayanan. Robust position-based routing in wireless ad hoc networks with unstable transmission ranges. In *DIALM '01: Proceedings of the 5th international workshop on Discrete algorithms and methods for mobile computing and communications*, pages 19–27, New York, NY, USA, 2001. ACM Press.

[3] Y. Ben-Asher, M. Feldman, and S. Feldman. Ad-hoc routing using virtual coordinates based on rooted trees. *SUTC*, 01:6–13, 2006.

[4] N. Carlsson and D. L. Eager. Non-euclidean geographic routing in wireless networks. *Ad Hoc Netw.*, 5(7):1173–1193, 2007.

[5] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: a decentralized network coordinate system. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architecture, and protocols for computer communications*, pages 15–26, New York, NY, USA, 2004. ACM Press.

[6] G. Finn. Routing and addressing problems in large metropolitan-scale internetworks. Technical Report ISI/RR-87-180, Mar. 1987.

[7] K. Gabriel and R. Sokal. A new statistical approach to geographic variation analysis. *Systematic Zoology*, 18:259–278, 1969.

[8] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on*

Information Theory, 46(2):388–404, 2000.

[9] D. B. Johnson, D. A. Maltz, and Y.-C. Hu. The dynamic source routing protocol for mobile ad hoc networks, 2003.

[10] B. Karp and H. T. Kung. Gpsr: greedy perimeter stateless routing for wireless networks. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 243–254, New York, NY, USA, 2000. ACM Press.

[11] S. K. Kaul, M. Gruteser, and I. Seskar. CRAWDAD trace set rutgers/noise/rssi (v. 2007-04-20). Downloaded from http://crawdad.cs.dartmouth.edu/rutgers/noise/RSSI, April 2007.

[12] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker. Geographic routing made practical. In *NSDI'05: Proceedings of the 2nd conference on the Symposium on Networked Systems Design & Implementation*, pages 16–16, Berkeley, CA, USA, 2005. USENIX Association.

[13] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker. On the pitfalls of geographic face routing. In *DIALM-POMC '05: Proceedings of the 2005 joint workshop on Foundations of mobile computing*, pages 34–43, New York, NY, USA, 2005. ACM Press.

[14] Y.-B. Ko and N. H. Vaidya. Location-aided routing (lar) in mobile ad hoc networks. *Wirel. Netw.*, 6(4):307–321, 2000.

[15] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger. Geometric ad-hoc routing: of theory and practice. In *PODC '03: Proceedings of the twenty-second annual symposium on the Principles of distributed computing*, pages 63–72, New York, NY, USA, 2003. ACM Press.

[16] F. Kuhn, R. Wattenhofer, and A. Zollinger. Asymptotically optimal geometric mobile ad-hoc routing. In *DIALM '02: Proceedings of the 6th international workshop on Discrete algorithms and methods for mobile computing and communications*, pages 24–33, New York, NY, USA, 2002. ACM Press.

[17] F. Kuhn, R. Wattenhofer, and A. Zollinger. Ad-hoc networks beyond unit disk graphs. In *DIALM-POMC '03: Proceedings of the 2003 joint workshop on the Foundations of mobile computing*, pages 69–78, New York, NY, USA, 2003. ACM Press.

[18] J. Li, J. Jannotti, D. S. J. De Couto, D. R. Karger, and R. Morris. A scalable location service for geographic ad hoc routing. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 120–130, New York, NY, USA, 2000. ACM Press.

[19] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc on-demand distance vector (aodv) routing, 2003.

[20] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica. Geographic routing without local information. In *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 96–108, New York, NY, USA, 2003. ACM Press.

[21] G. Toussaint. The relative neighborhood graph of a finite planar set. *Pattern Recognition*, 12(4):261–268, 1980.